Table of Contents

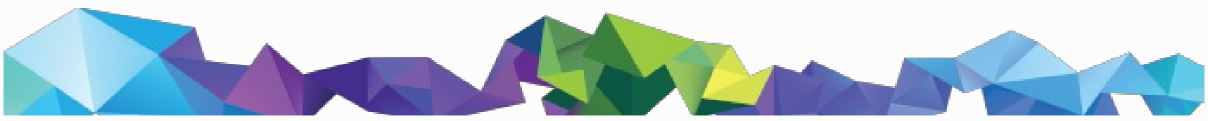
Job Manager for local execution of QuantumATK scripts

Version: T-2022.03

Downloads & Links

- PDF version
- PDF version for QuantumATK 2016
- silicon.py
- cnt.py
- mpi_check.py

Table of Contents:

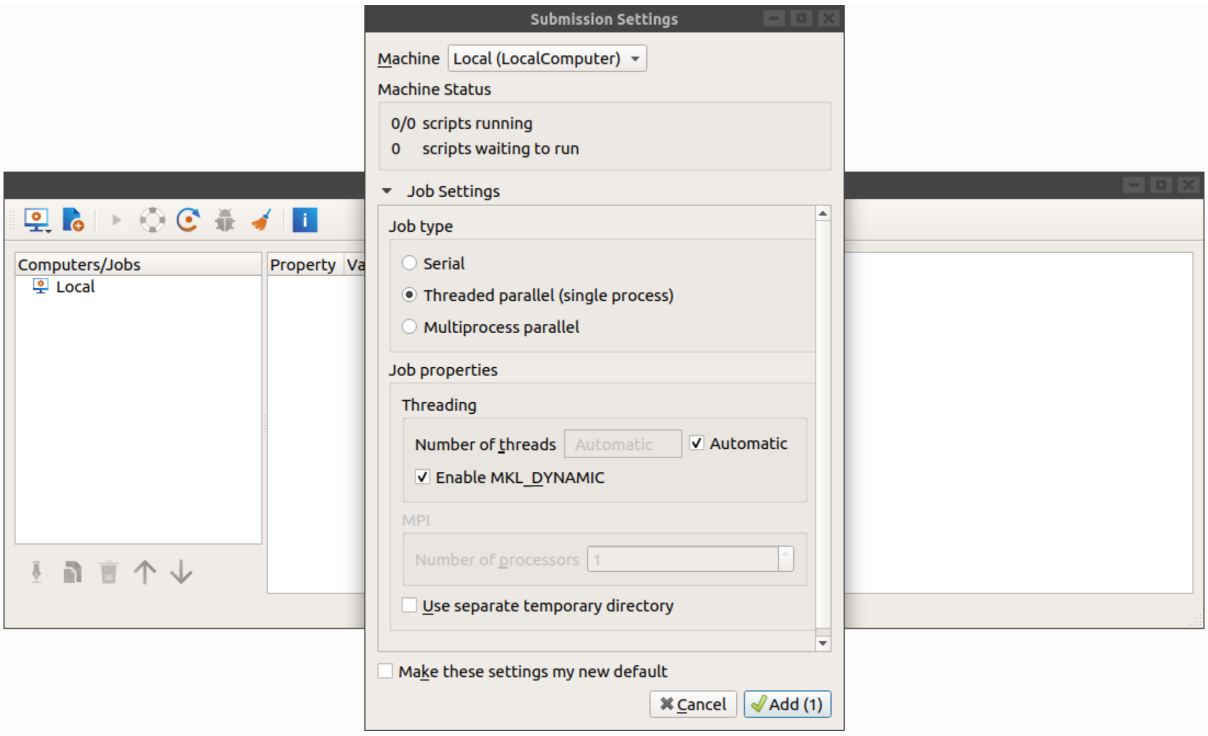


Execute QuantumATK simulations via the Job Manager

In this section you will learn how to use the **Job Manager** for local execution of QuantumATK scripts. Specifically, you will learn about queuing, running and managing QuantumATK jobs.

Create a new empty project and download the example script silicon.py, which runs an ATK-DFT calculation with very many k-points (31x31x31).

Drop it on the **Job Manager** and select a local computer for the job execution.


Submission Settings

Machine: Local (LocalComputer)

Machine Status

0/0 scripts running
0 scripts waiting to run

Job Settings

Job type

( ) Serial
(●) Threaded parallel (single process)
( ) Multiprocess parallel

Job properties

Threading

Number of threads: Automatic [✓] Automatic
[✓] Enable MKL_DYNAMIC

MPI

Number of processors: 1

[ ] Use separate temporary directory

[ ] Make these settings my new default

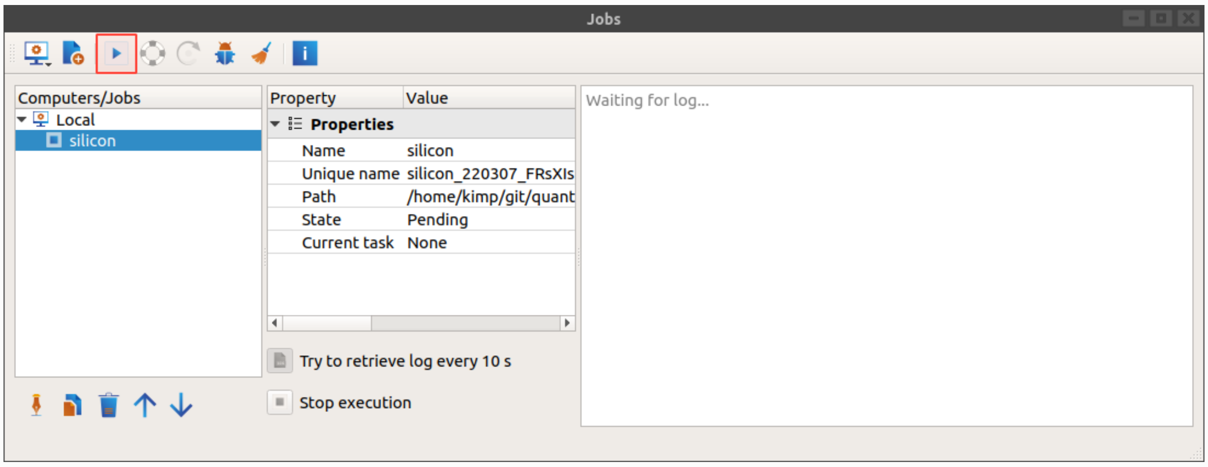
The **Submission Settings** widget has two basic panels:

- **Job type;**
  - Serial
  - Threaded parallel (single process)
  - Multiprocess parallel

- **Job properties;**
  - Threading;
  - MPI Parallelism;
  - **Use separate temporary directory** (set this to allow using a non-default working directory for running the job. Note, results will not be appended to existing hdf5 or nc files, but instead a new file will be created.)

Set these settings according to your needs, and click **Add**.

In the **Job Manager**, first select the job. Then click the highlighted **Run** ▶ button to start the job.

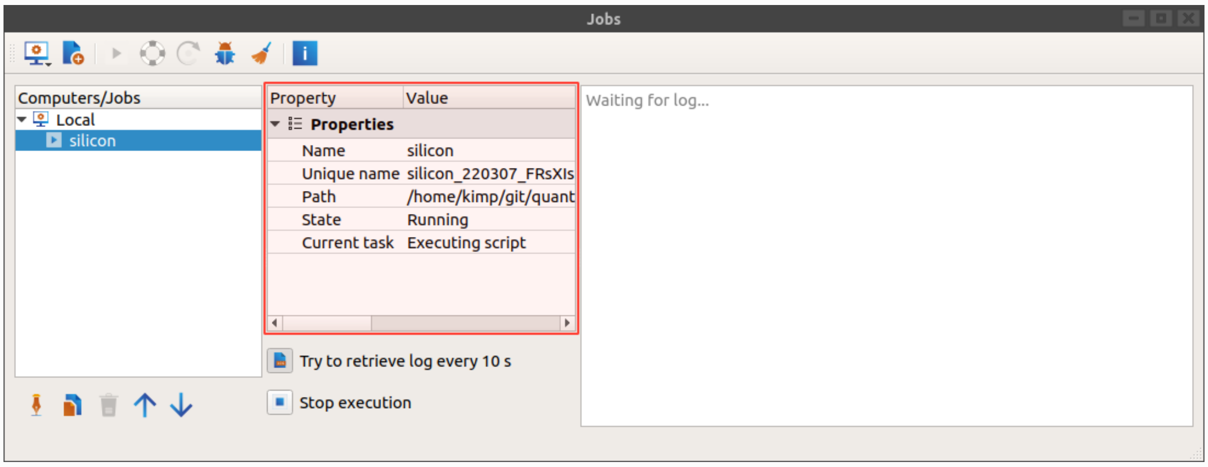

Jobs

Computers/Jobs
- Local
  - silicon

Property	Value
Properties	
Name	silicon
Unique name	silicon_220307_FRsXIs
Path	/home/kimp/git/quant
State	Pending
Current task	None

Waiting for log...

The task state changes from “Pending” to “Running”.


Jobs

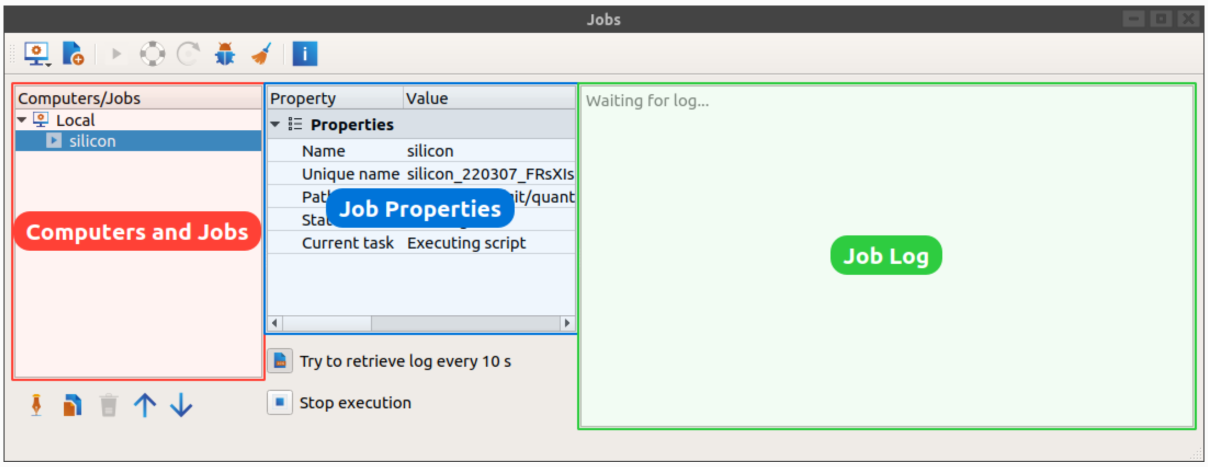
Computers/Jobs
Local
silicon

Property	Value
Properties	
Name	silicon
Unique name	silicon_220307_FRsXIs
Path	/home/kimp/git/quant
State	Running
Current task	Executing script

Try to retrieve log every 10 s
Stop execution

Waiting for log...

The Job Manager has three main parts. The overview of **computers and jobs** highlighted in red, the **properties of the job highlighted** in blue, and **the log** highlighted in green.



The properties of the job is a Property–Value list which shows all details of the settings used for job execution, including:

- Name of the Python script.
- State.
- Current task.

The job finishes after about 1 minute (2.5 GHz CPU). Note that the task state changes to “Finished”.

The job output (calculation results) of course appears in the QuantumATK **DataView** after job execution.

Note

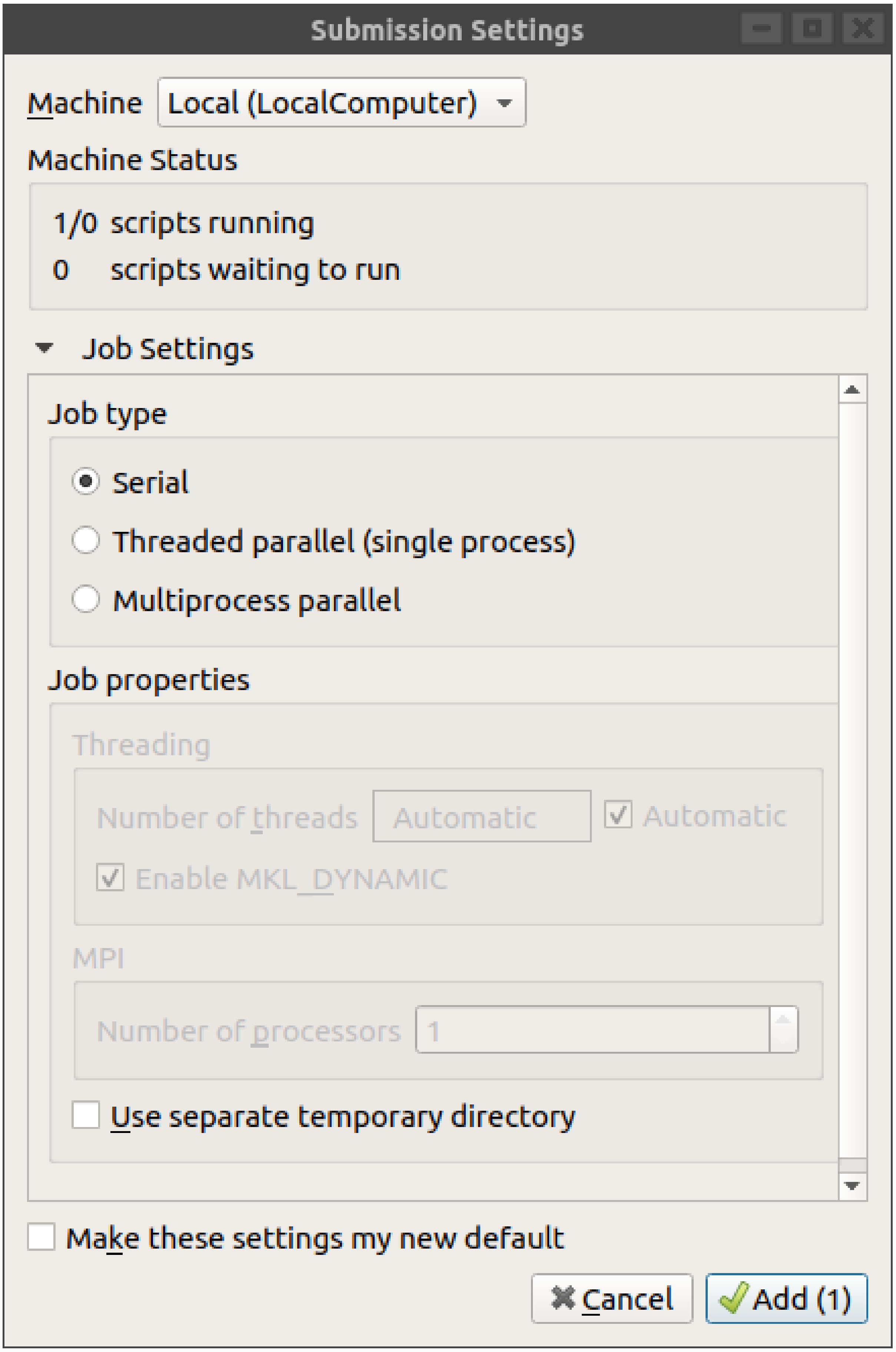
Remember that the default job type is “Threaded parallel (Single process)”. You can change this to “Serial” or “Multiprocess parallel” before starting the job.

Use the **Trash** button to remove jobs from the job queue.

Serial execution

In the **Submission Settings** window select a **Serial** job type as shown in the below figure to run on a single process with no threading.

In fact, note that threading and MPI parallelization are not available.



Submission Settings

Machine: Local (LocalComputer)

Machine Status

1/0 scripts running
0 scripts waiting to run

Job Settings

Job type

- [● Serial]
- ○ Threaded parallel (single process)
- ○ Multiprocess parallel

Job properties

Threading

Number of threads: Automatic ☑ Automatic

☑ Enable MKL_DYNAMIC

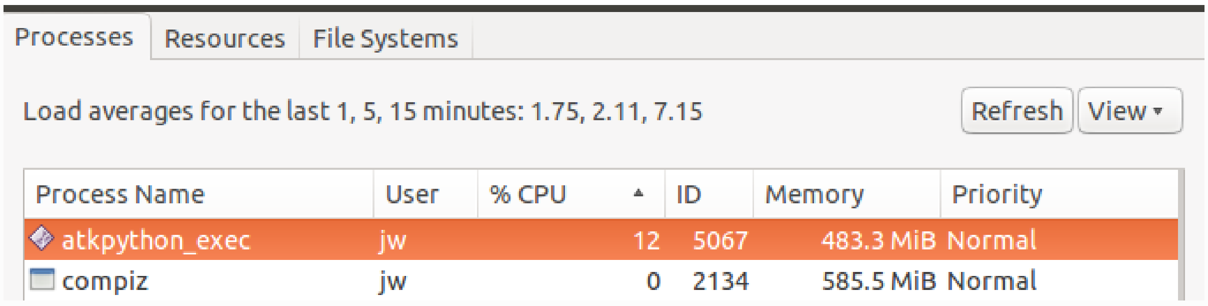
MPI

Number of processors: 1

☐ Use separate temporary directory

☐ Make these settings my new default

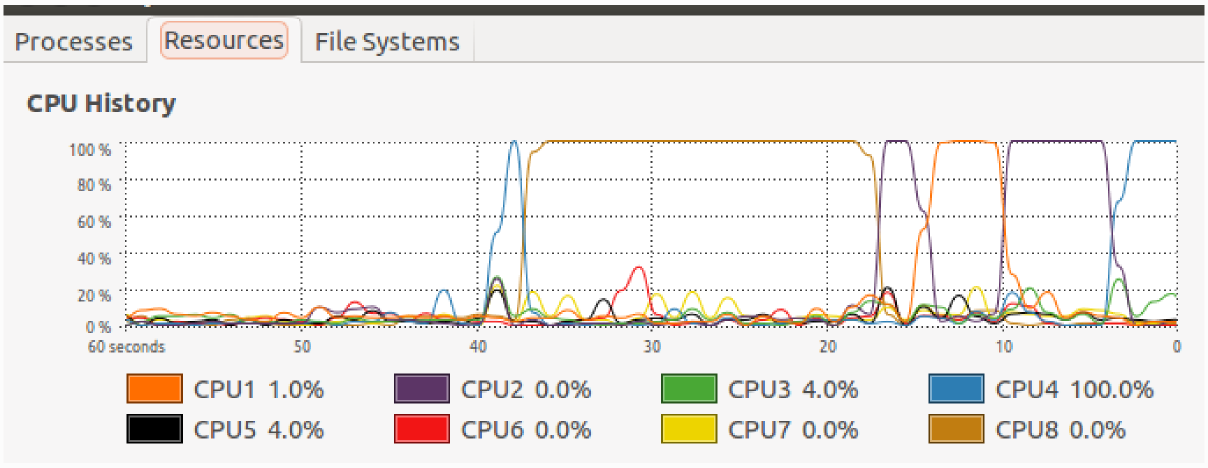
If you check the system load during local execution in serial, you should see that the serial job launches only a single computing task on a single CPU core.



Load averages for the last 1, 5, 15 minutes: 1.75, 2.11, 7.15

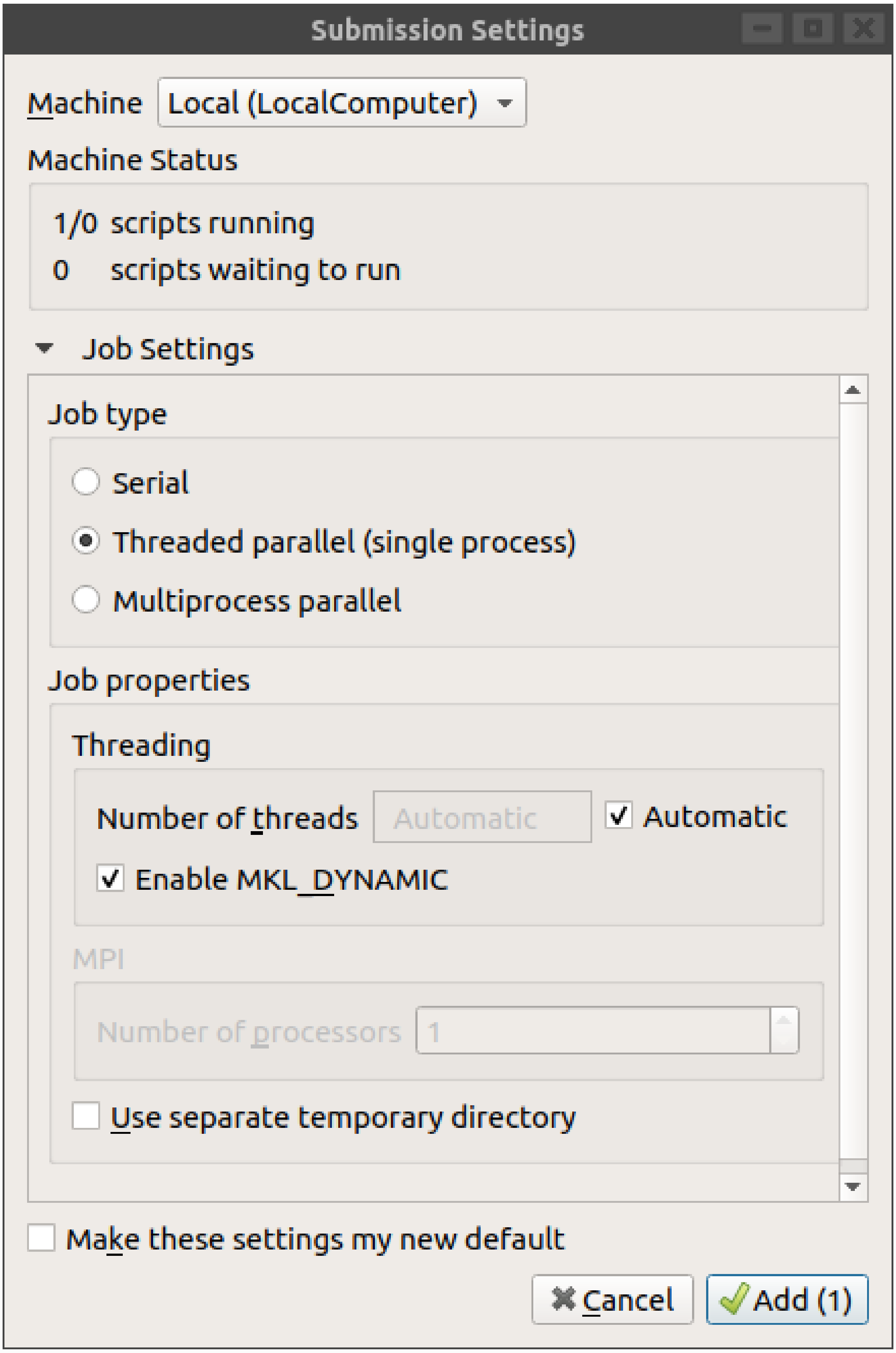
Process Name	User	% CPU	ID	Memory	Priority
atkpython_exec	jw	12	5067	483.3 MiB	Normal
compiz	jw	0	2134	585.5 MiB	Normal

Only one core is used at a time, but the hardware process manager may move the task between cores from time to time.



Threading

In the **Submission Settings** window select a **Threaded parallel (single process)** job type as shown in the below figure to run on a single process with threading.



Submission Settings

Machine: Local (LocalComputer)

Machine Status

1/0 scripts running
0 scripts waiting to run

Job Settings

Job type

( ) Serial
(●) Threaded parallel (single process)
( ) Multiprocess parallel

Job properties

Threading

Number of threads: Automatic [✓] Automatic

[✓] Enable MKL_DYNAMIC

MPI

Number of processors: 1

[ ] Use separate temporary directory

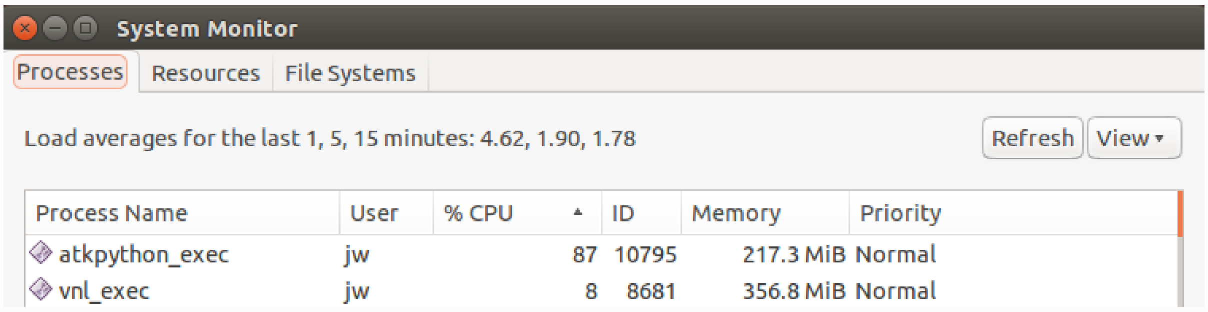
[ ] Make these settings my new default

Threading is one way to parallelize a computational job. QuantumATK uses Intel MKL for openMP threading. Note that we do in general recommend MPI parallelization over threading for parallelizing DFT calculations. However, threading is often more efficient for parallelizing **ATK-ForceField** calculations, and Hybrid DFT calculations.

Download the script cnt.py, which uses ATK-ForceField to calculate the dynamical matrix of a multi-wall carbon nanotube.

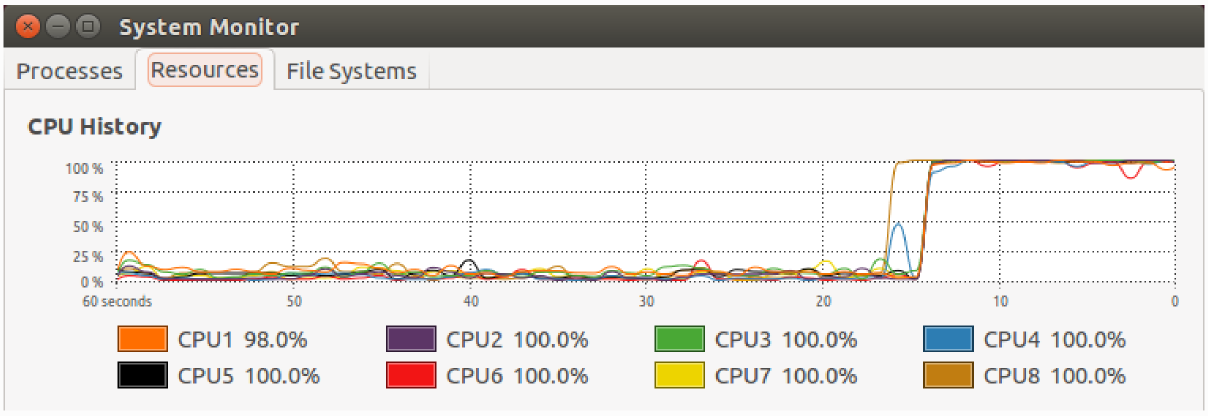
Execute it using the **Job Manager**, and choose job type “Threaded parallel (single process)”. It should be pretty fast.

If you check the system load during execution of the calculation, you should see that only a single **atkpython** process is started, even though several cores appear to be busy. This is because the work load of the one process is split into a number of threads that may be distributed on more cores.



Load averages for the last 1, 5, 15 minutes: 4.62, 1.90, 1.78

Process Name	User	% CPU	ID	Memory	Priority
atkpython_exec	jw	87	10795	217.3 MiB	Normal
vnl_exec	jw	8	8681	356.8 MiB	Normal



Download the script cnt.py to test the performance of a ATK-ForceField simulations using threading. This specific example will calculate the dynamical matrix of a multi-wall carbon nanotube. If you also run the calculation in serial, you will see that the wall-clock time used for evaluating ATK-ForceField forces may decrease significantly when threading is switched on. In the example shown below, the time spent on force calculations is roughly halved.

```
==> cnt_threading.log <==
Timing:
-----------------------------------------------------------------------
Forces                    :     14.22 s       0.00 s      60.63% |============|
Loading Modules + MPI     :      1.57 s       1.57 s       6.69% |=|
File IO, nlsave           :      0.10 s       0.05 s       0.43% |
-----------------------------------------------------------------------

==> cnt_serial.log <==
Timing:
-----------------------------------------------------------------------
Forces                    :     22.37 s       0.01 s      79.53% |============|
Loading Modules + MPI     :      1.55 s       1.55 s       5.52% ||
File IO, nlsave           :      0.06 s       0.03 s       0.21% |
-----------------------------------------------------------------------
```

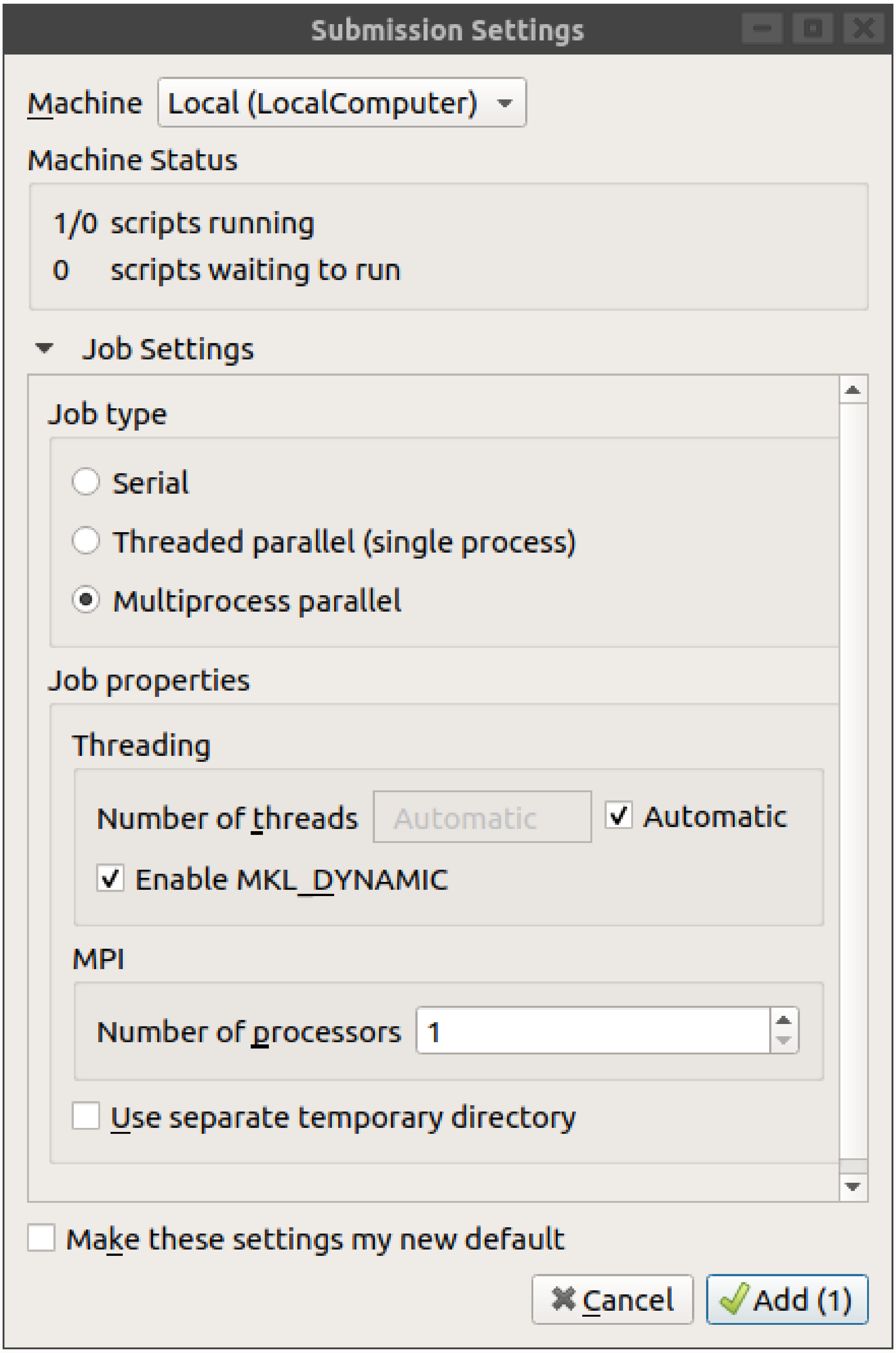
MPI parallelization

Important

If you are running **ATK 2016 or earlier** you need to have MPI installed on your local computer.

Both the **Linux** and **Windows** versions are compiled against Intel MPI library. Since **ATK 2017** Intel’s `mpiexec.hydra` is provided on both Windows and Linux versions - **this is the recommended way to run QuantumATK in parallel**.

In the **Submission Settings** window select a *Multiprocess parallel* and e.g. 4 MPI processes.


Submission Settings

Machine: Local (LocalComputer)

Machine Status

1/0 scripts running
0 scripts waiting to run

Job Settings

Job type

- Serial
- Threaded parallel (single process)
- [✓ Multiprocess parallel]

Job properties

Threading

Number of threads: Automatic [✓ Automatic]

[✓ Enable MKL_DYNAMIC]

MPI

Number of processors: 1

☐ Use separate temporary directory

☐ Make these settings my new default

Running from the command line

If you wish to run QuantumATK in parallel from the command line you can use the `mpiexec.hydra` binary shipped with QuantumATK and located in the folder `libexec/mpiexec.hydra` present in your installation folder.

In this case you can run parallel jobs with:

```
$ QW_INSTALLATION_PATH/libexec/mpiexec.hydra -n 4 atkpython atk_script.py
```

Hint

Prepend `QW_INSTALLATION_PATH/libexec/` in your `PATH`.

```
$ export PATH=QW_INSTALLATION_PATH/libexec:$PATH
```

This way you will automatically pick up the `mpiexec.hydra` binary shipped with QuantumATK:

```
$ mpiexec.hydra -n 4 atkpython atk_script.py
```

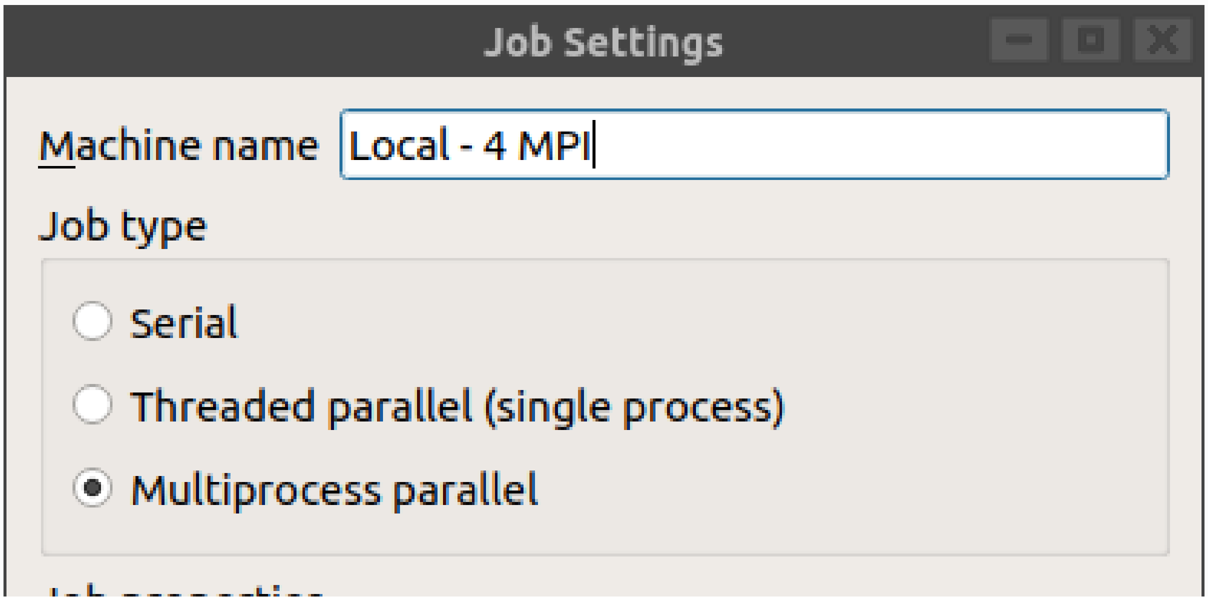
Adding Computers

It may sometimes be convenient to have a predefined local computer that is set up with MPI parallelization as default mode. You can easily add such a computer yourself.

In the Job Manager main window, click to add a new computer and select New Local.

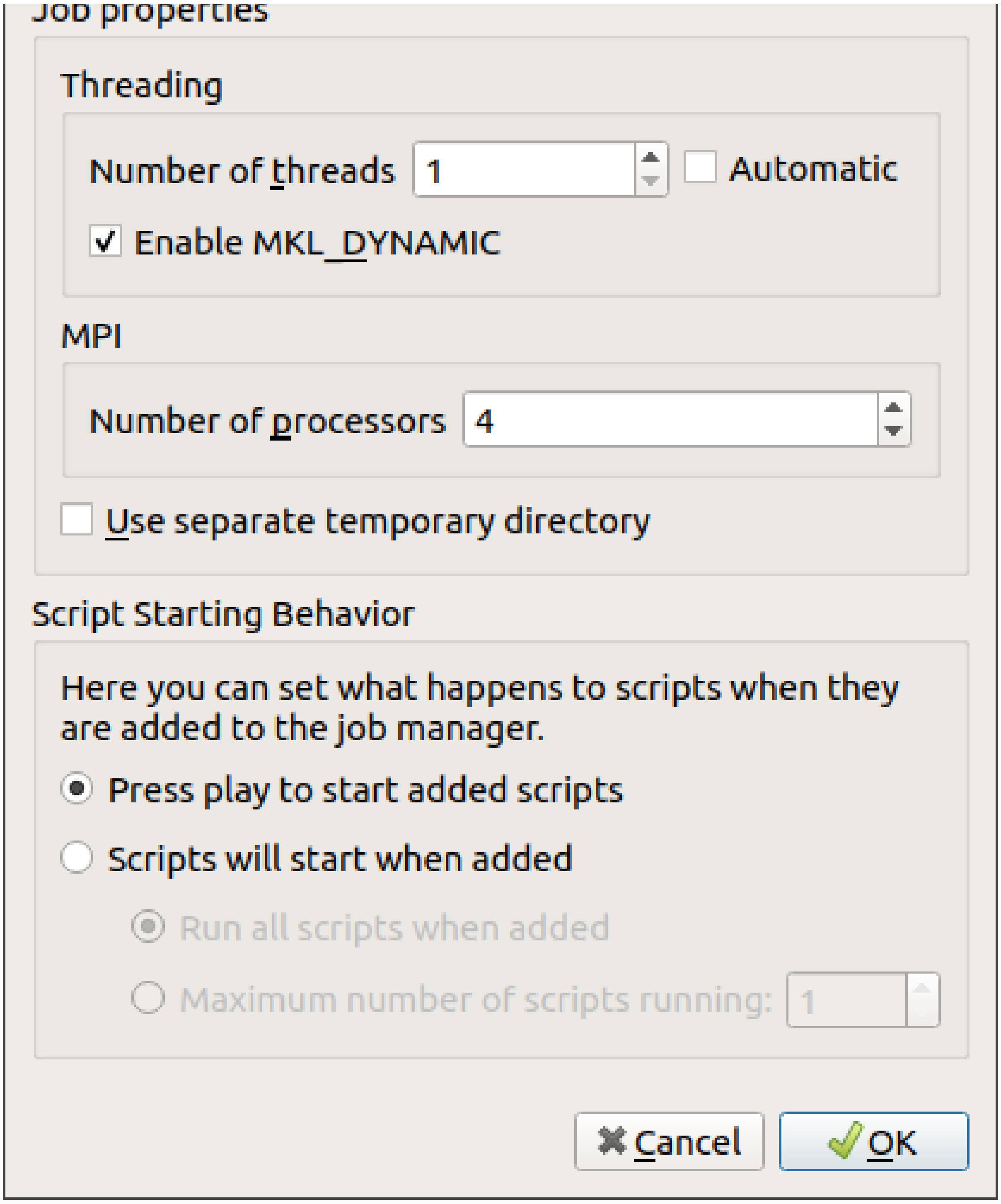
Then edit the default job settings of the new computer in the **Job Settings** window that pops up:

- Name the computer, e.g. “Local - 4 MPI”.
- Select *Multiprocess Parallel* as job type.
- Make sure threading is turned off (*Number of threads* = 1)
- Choose the default number of processors, e.g. 4.
- Click *OK* to add the new computer.


Job Settings

Machine name: Local - 4 MPI

Job type
- Serial
- Threaded parallel (single process)
- [✓ Multiprocess parallel]


Job properties

Threading

Number of threads: 1 [ ] Automatic

[✓] Enable MKL_DYNAMIC

MPI

Number of processors: 4

[ ] Use separate temporary directory

Script Starting Behavior

Here you can set what happens to scripts when they are added to the job manager.

[● Press play to start added scripts]

○ Scripts will start when added

[● Run all scripts when added]

○ Maximum number of scripts running: 1

You can add as many custom computers as you like.

The **Job Settings** widget has an additional panel in addition to *Job type* and *Job properties*:

There are two options. Either you manually press play to run the jobs, or you can automatically start jobs submitted to the computer.

Set these settings according to your needs, and click **Add**.

Previous Next